Received 31 March 2023, accepted 16 May 2023, date of publication 22 May 2023, date of current version 6 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3278837

RESEARCH ARTICLE

Multi-Feature and Multi-Modal Mispronunciation Detection and Diagnosis Method Based on the Squeezeformer Encoder

SHEN GUO[1,2], **ZAOKERE KADEER**[1,2], **AISHAN WUMAIER**[1,2], **(Member, IEEE), LIEJUN WANG**[2], **AND CONG FAN**[1,2]

[1]Key Laboratory of Multilingual Information Technology, Ürümqi 830046, China
[2]School of Information Science and Engineering, Xinjiang University, Ürümqi 830046, China

Corresponding authors: Zaokere Kadeer (zuhra@xju.edu.cn) and Aishan Wumaier (hasan1479@xju.edu.cn)

This work was supported by the National Natural Science Foundation Project under Grant U1903213.

ABSTRACT In recent years, with the development of deep learning, research on end-to-end mispronunciation detection and diagnosis(MDD) methods has been further promoted. At present, research on end-to-end mispronunciation detection and diagnosis is gradually emerging. Most end-to-end mispronunciation detection and diagnosis methods are based on the CNN-RNN-CTC network structure. To improve the performance of end-to-end mispronunciation detection and diagnosis systems, this paper proposes an end-to-end multi-feature and multi-modal mispronunciation detection and diagnosis method based on the Squeezeformer encoder. The model uses Squeezeformer as an audio encoder, a Bi-LSTM network as a phoneme encoder, and Transformer as a decoder. The model fuses phoneme information before speech encoding and decoding, respectively, and uses a secondary decoding mechanism during the decoding process. This study further incorporated phoneme information in the encoding process so that the model could learn the intrinsic characteristics of the speaker's pronunciation content. The decoding process uses a secondary decoding mechanism to send the sequence decoded by the model to the decoder for decoding again, which solves the problem of no a priori knowledge at the decoder end in the first decoding stage, thus improving the performance of mispronunciation detection and diagnosis. In this study, experiments were conducted on the PSC-Reading Mandarin mispronunciation detection and diagnosis dataset. Compared with the baseline model, the F1 index improved from 0.4060 to 0.7943, and the diagnostic accuracy improved from 83.93% to 88.45%.

INDEX TERMS MDD, squeezeformer, secondary decoding, computer-assisted language learning (CALL).

I. INTRODUCTION

With the development of deep learning technology, the technology in the field of speech recognition has been pushed to a new level. Deep learning models such as Transformer [1], Conformer [2], and Squeezeformer [3] have been proposed by researchers and applied in the field of speech recognition [4] or natural language processing [5], with a more complex structure and stronger feature information representation ability than traditional deep neural networks, which greatly improve the performance of speech recognition or natural language processing. End-to-end mispronunciation detection and diagnosis is performed with phoneme recognition [6] as the task, and the phoneme sequences identified by the model and the reference phoneme sequences are aligned by the Needleman–Wunsch [7] algorithm, which leads to the final mispronunciation detection and diagnosis [8] results. The traditional approach is used to determine whether pronunciation is correct or not based on confidence scores, but this approach can only test whether pronunciation is correct or not, and cannot provide a valid diagnosis of pronunciation. The confidence-based mispronunciation detection technique relies on more algorithm modules, with a large degree of coupling between modules; the training

The associate editor coordinating the review of this manuscript and approving it for publication was Khoa Luu.

process of this model is more complicated, which affects the final mispronunciation detection performance. To assess the types of mispronunciation and to provide feedback about specific errors, the extended recognition network (ERN) [9] was created. The extended recognition network incorporates the expected mispronunciation patterns into the lexicon to constrain the recognition path to canonical pronunciations and possible mispronunciations. These extended recognition networks constructed from manual or data-driven rules have the advantage of simultaneously detecting errors and providing diagnosis, and thus can systematically provide diagnostic feedback. However, it is difficult to construct extended recognition networks that contain as many mispronunciation paths as possible, thus limiting their performance in mispronunciation detection and diagnosis. Leung proposed a CNN-RNN-CTC [10] model for mispronunciation detection and diagnosis tasks, using a connectionist temporal classification (CTC) [11], [12] loss function instead of a cross-entropy loss function to train the model and remove the reliance on phoneme boundary information during training. The training process of the mispronunciation detection and diagnosis model can be further improved by using a multi-task learning model with multiple encoders to fuse linguistic information to improve the model's ability to carry out phoneme recognition, thus enhancing the performance of mispronunciation detection and diagnosis. Feng proposed the SED-MDD [13] model, which further improves the performance of mispronunciation detection and diagnosis based on the CNN-RNN-CTC model by adding a text encoder to the end-to-end phoneme recognition model and fusing acoustic and linguistic information to achieve a reference text-based mispronunciation detection and diagnosis method. Yunfei Shen proposed a mispronunciation detection and diagnosis model with a WavLM-Transformer [14] structure, using Transformer instead of the traditional CNN-RNN structure in the encoding stage, using a self-supervised pre-training model to extract raw audio features for training, and using a hybrid CTC/ATT structure as the loss function to further improve the performance of the mispronunciation detection and diagnosis model.

In this paper, we propose a multi-feature, multi-modal mispronunciation detection and diagnosis model with Squeezeformer as the audio encoder and Transformer as the decoder, which incorporates phoneme length information into speech features before speech encoding, and uses a Long Short Term Memory networks (LSTM) [15] to encode phoneme sequences in the phoneme encoding stage, and uses an attention mechanism to further fuse audio features and phoneme features so that the model can learn the speaker's pronunciation patterns. The decoding process uses a secondary decoding mechanism, which further improves the performance of mispronunciation detection and diagnosis. In this paper, the WavLM-Transformer was used as the baseline model; the F1 index improved from 0.4060 to 0.7943, and the diagnostic accuracy improved from 83.93% to 88.45% when compared with the baseline system.

II. RELATED WORKS ON MDD

This section introduces the development process of MDD, including from traditional methods to deep learning methods, and the research background of Squeezeformer.

A. STATISTICAL APPROACHES

A speech recognition system based on statistical learning algorithms is essentially a statistical model of the pronunciation patterns of various phonemes in a language, so the likelihood of the output of a speech recognition system trained using a standard speech dataset for a segment of speech can be used to measure the similarity between that segment of speech and the standard speech. The speech-recognition-based approach views the detection of articulation errors as a measure of how well a segment of speech can be correctly recognized by a standard speech recognition system, that is, the confidence score of decoding a signal into a target phoneme pattern in a speech recognition system [16]. This requires the construction of efficient and reasonable confidence measures that can effectively distinguish correct pronunciation from mispronunciation.

In the 1990s, SRI International conducted a series of studies on automatic pronunciation evaluation, and Bernstein proposed the use of HMM [17]-based speech recognition models with Viterbi [18] algorithms for forced alignment between audio and recognized phoneme sequences and for pronunciation quality evaluation. Neumeyer, on the other hand,proposed an algorithm to calculate pronunciation scores using the log-likelihood output from the decoding of HMM speech recognition systems. Based on this, Witt proposed the GOP [19] algorithm and described a system for pronunciation evaluation based on forced alignment and the GOP algorithm, in which a phoneme is considered to have poor pronunciation quality when its GOP score is below a predefined threshold. In this regard, the framework of pronunciation quality assessment based on a speech recognition system with the GOP algorithm is determined, and the main direction of further research is to improve the defects of the GOP algorithm and to improve the correlation between GOP machine scores and expert-labeled artificial scores.

Ke Yan proposed a trainable posterior probability transformation of phoneme correlation to fit the artificial scores by Sigmoid transformation, which significantly improved the human–machine score correlation coefficient of the GOP algorithm. Novoa et al. proposed an improved GOP algorithm considering the HMM transfer probability in the DNN-HMM [20] acoustic model. After the emergence of deep learning algorithms, some improved GOP algorithms also used deep learning models. Shi et al. proposed the context-dependent CaGOP algorithm, which predicts the duration of each phoneme by feeding the reference text into a self-attentive text-based encoder during GOP calculation, and uses the difference between the predicted duration and the actual duration of the phoneme obtained by forced alignment as the penalty factor in GOP computation [21].

In response to the shortcoming of assuming equal phoneme prior probabilities in the GOP algorithm, Long Zhang fused the phoneme confusion prior knowledge into the calculation of phoneme prior probabilities in the GOP calculation and used the confusion phoneme set instead of the full probability space to improve the human–computer score correlation coefficient of Mandarin vowel-rhyme assessment from 0.796 to 0.836 in the baseline GOP algorithm. Since Witt proposed the pronunciation fit scoring model in 2000, the pronunciation fit scoring algorithm based on the GMM-HMM recognition model has been widely used in mispronunciation detection applications. In recent years, Hu and Ming have carried out some optimization of the pronunciation fit scoring algorithm. This technical route of the algorithm for mispronunciation detection has advantages such as ease of construction (the ability to migrate directly from recognition models) and low data annotation requirements (no phoneme-level annotation data required). At the same time, the method has some limitations, such as low accuracy and no ability to diagnose mispronunciations. Professor Helen Meng proposed the Extended Speech Recognition Network (ERN) and has continued to improve the method in their subsequent research work. The aim of the ERN approach is to detect and diagnose mispronunciations by manually developing a series of rules for mispronunciations to be added to the standard speech recognition network. Based on the popular deep neural networks, Professor Helen Meng proposed the phoneme-level acoustic model [22] (APM) algorithm for detecting and diagnosing mispronunciations and established a series of widely used evaluation metrics. A phoneme transcription alignment model is first trained for each frame of the audio, based on which a recognition model is trained based on the contextual features and transcription alignment results for each frame of that audio, and, finally, a Viterbi decoder is used to obtain the final results.

B. END-TO-END MDD METHODS

With the development of deep learning in recent years, mispronunciation detection and diagnosis algorithms based on deep neural networks or end-to-end [23] speech recognition models have become a hot research topic, and some end-to-end mispronunciation detection and diagnosis models have also emerged. These models no longer need HMM models and have gradually removed forced alignment from the training process.

Watanabe et al. proposed an end-to-end speech evaluation system based on the CTC/Attention [24] end-to-end speech recognition model and discussed the effect of fundamental frequency features on the performance of an end-to-end speech recognition system for Mandarin, which compares the recognition results of the reference text and the speech recognition system using the Needleman–Wunsch algorithm to obtain mispronunciation detection and diagnosis results. Leung et al. proposed a CNN-RNN-CTC [10] model for mispronunciation detection and diagnosis tasks, using a CTC loss function instead of a cross-entropy loss function to train the model, thus eliminating the need to provide phoneme boundary information at training time.

Feng et al. proposed the SED-MDD [13] algorithm, which implements an end-to-end pronunciation evaluation algorithm related to the reference text by adding a text encoder to an end-to-end phoneme recognition model of Encoder-Attention-Decoder architecture. Based on SED-MDD, Fu et al. [25] combines the reference text encoder with a CNN-RNN-CTC model, and implicitly aligns the phoneme sequences from the audio to be evaluated and the reference text, respectively, inside the model using an attention mechanism. Zhang et al. proposed a text-related mispronunciation detection and diagnosis model based on Transformer [26], which performs alignment between the actual transcript and the reference text before training to obtain the corresponding mispronunciation, thus defining the mispronunciation detection and diagnosis task as a binary classification task of ‘‘correct or incorrect pronunciation’’, and further improving the overall performance of the model by adding accent recognition and phoneme recognition tasks. The speech evaluation system proposed by Nadig et al. takes uncertainty into account by concatenating an Encoder-Attention-Decoder [27]-based pronunciation evaluation model after a CTC-based phoneme recognition model so that a pronunciation score can be given simultaneously after aligning the recognition result with the reference text.

Shen et al. [14] proposed a mispronunciation detection and diagnosis model with a WavLM-Transformer structure, using Transformer instead of the CNN-RNN structure in the encoding stage, using a self-supervised pre-training model to extract audio raw features for training, and using a hybrid CTC/ATT structure as the loss function to further improve the performance of the mispronunciation detection and diagnosis model. However, the model does not perform best on a series of indicators such as F1 value and diagnostic accuracy.

C. SQUEEZEFORMER SPEECH REPRESENTATION MODEL

With the further development of deep learning, the Transformer model has been proposed in the field of machine translation [28] as a new deep learning algorithmic framework that has received more and more attention from researchers. The self-attention mechanism in the Transformer model [29] is inspired by the fact that humans focus only on what is important and learn only the important information in the input sequence. Transformer can integrate acoustic, articulatory, and language models [30] into a single neural network to form an end-to-end speech recognition system, solving the problems of the forced alignment and multi-module training of traditional speech recognition systems. However, models with self-focus or convolution [31] each have their limitations. While Transformer is good at modeling remote global contexts, it is weak at extracting fine-grained local feature patterns. As a result, a convolutional enhancement Transformer for speech recognition, named Conformer, was later proposed. Conformer significantly outperforms previous Transformer- and CNN-based models, with local and global information extraction acting together to learn location-dependent local features and use content-based global interactions to achieve better accuracy.

However, through a series of systematic studies, the researchers found that the design choice of the Conformer architecture was not optimal. After re-examining the design choices of macro- and microarchitectures for Conformer, the researchers proposed Squeezeformer, which consistently outperformed the Conformer model for the same training scenario. In particular, for the macroarchitecture, Squeezeformer combines the temporal U-Net [32] structure, which reduces the cost of multi-headed attention [33] modules on long sequences, with a simpler block structure of multi-headed attention or convolution modules followed by feedforward modules, instead of the macaron structure proposed in Conformer. Moreover, for the microarchitecture, Squeezeformer simplifies the activation in the convolution block, removes redundant layer normalization operations, and merges effective depth downsampling layers to efficiently subsample the input signal, leading to further performance improvements.The Squezeformer model structure is shown in Fig.1.

III. OUR METHODS

A. SYSTEM OVERVIEW

The input of the model is the fbank acoustic feature, phoneme sequence length information embedding, and phoneme sequence embedding, and the output of the model is the corresponding acoustic phoneme sequence. The mispronunciation detection and diagnosis model proposed in this paper uses the Squeezeformer model in the encoding stage, uses the Bi-LSTM network to encode the reference phoneme sequence, and fuses the phoneme sequence length information before speech encoding. Our phoneme sequence uses a 512-dimensional embedding layer, and phoneme length information uses an 80-dimensional embedding layer. The audio encoding stage of the model uses 12 encoder layers, the number of multi-head attention heads is 4, and the phoneme encoder uses 1 layer of the Bi-LSTM network. The decoder of the model uses 2048 linear units and 6 encoder layers. The number of attention heads with multiple heads is 4. The loss function uses CTC/Attention as the joint loss. The overall structure of the model is shown in Fig.2.

The training and decoding process of the model is as follows:

- Convert audio into fbank features through calculation;
- Encode the length information of the phoneme sequence label into a vector through an embedding operation;
- Send the multi-modal feature obtained by fusing the fbank feature and the phoneme embedding vector to the Squeezeformer encoder;
- Encode the reference phoneme sequence through the LSTM network;
- Fuse the hidden state sequence calculated and output by the Squeezeformer encoder and the reference phoneme hidden state sequence encoded by the LSTM network through the attention mechanism;
- Input the multi-feature hidden state sequence into the Transformer decoder to calculate the joint loss, and then conduct backpropagation to optimize the network parameters;
- Use the trained model to decode the data to obtain the predicted phoneme sequence;
- Align the decoded phoneme sequence with the reference phoneme sequence to obtain the final mispronunciation detection and diagnosis results.

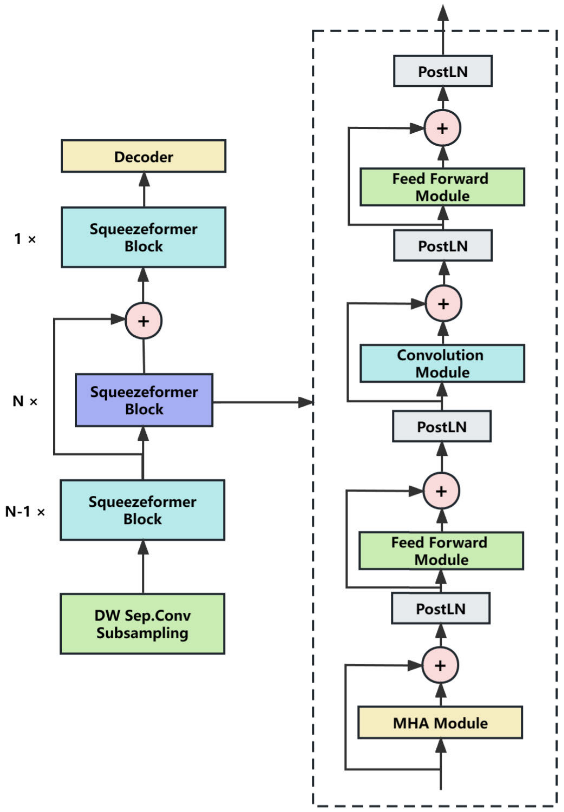

FIGURE 1. The Squeezeformer encoder.

B. MULTI-MODAL INFORMATION FUSION

Before sending the feature into the audio encoder, we fuse the length information of the phoneme tag sequence and the audio feature to form a multi-modal feature and then feed the multi-modal feature to the audio encoder. The multi-modal feature contains the length information of the phoneme tag sequence so that the model can further learn some intrinsic characteristics of the speaker's pronunciation content. First, we calculate the length of the corresponding phoneme tag sequence for each audio and store the phoneme tag sequence length and the corresponding audio name in the form of a form. During the model training process, the phoneme tag sequence length information is included in each batch of data. We embed the length information of the phoneme sequence label into an 80-dimensional vector $\mathbf{Z} \in R^{B \times 1 \times 80}$

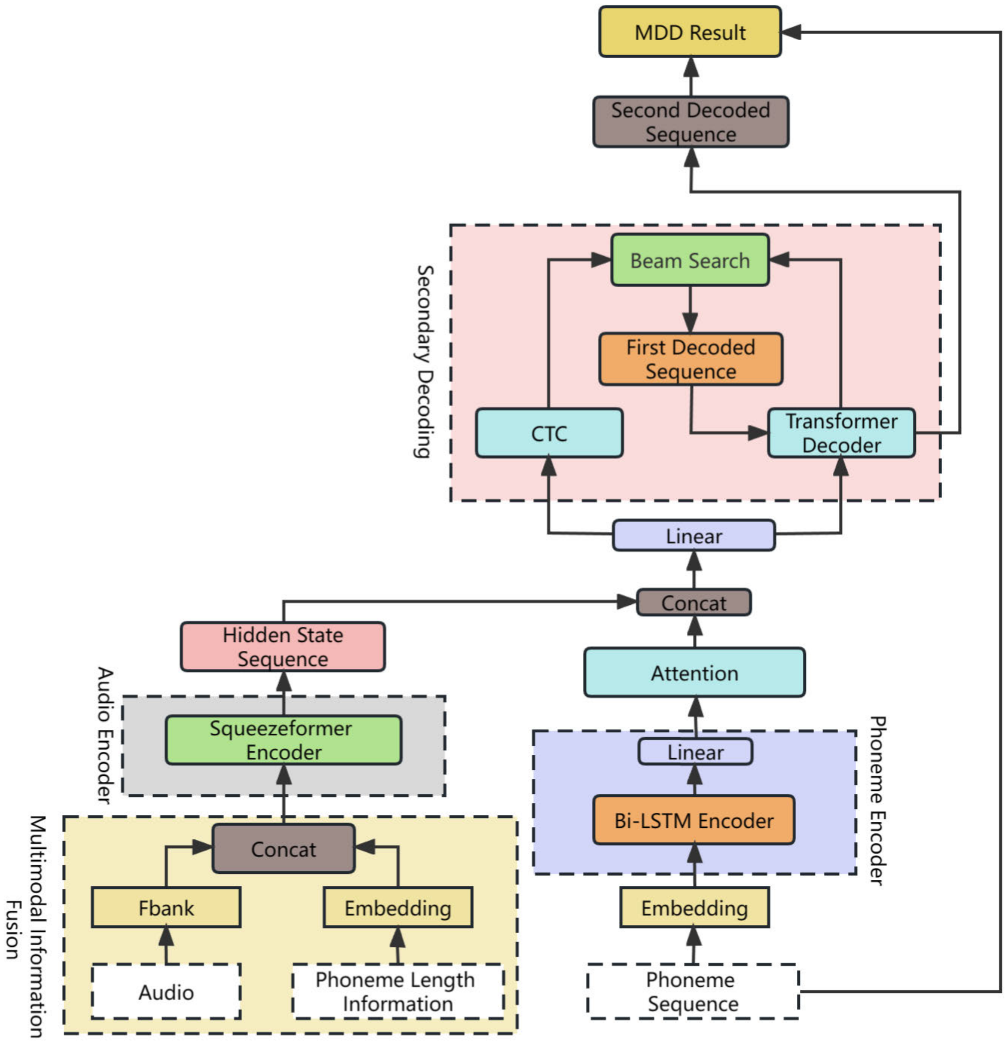

FIGURE 2. MDD model based on squeezeformer encoder.

through the embedding operation and then splice it with the audio feature $\mathbf{X} \in R^{B\times T\times 80}$ to obtain the multi-modal feature $\mathbf{Y} \in R^{B\times (T+1)\times 80}$. The whole multi-modal information construction method is shown in Fig.3.

C. AUDIO ENCODER

The audio encoding calculation process is as follows. The input of the audio encoder is the multi-modal feature after 80-dimensional phoneme information (phoneme length) embedding and 80-dimensional fbank feature splicing. First, the fbank feature of audio data is extracted, and the fbank feature of voice data is marked as $\mathbf{X} = [\mathbf{x_1}, \ldots \mathbf{x_n}]$. Phoneme information is embedded as $\mathbf{Z} = [\mathbf{z_1}, \ldots \mathbf{z_n}]$. Features after splicing $\mathbf{Y} = [\mathbf{y_1}, \ldots \mathbf{y_n}]$ output the corresponding hidden state sequence through the Squeezeformer encoder and mark it as $\mathbf{H^Q} = [\mathbf{h_1^q}, \ldots \mathbf{h_n^q}]$, so the audio encoder based on Squeezeformer can be expressed as

$$\mathbf{Z} = \text{Embedding}(\text{PhonemeLength}) \tag{1}$$

$$\mathbf{Y} = \text{Concat}(\mathbf{X}, \mathbf{Z}) \tag{2}$$

$$\mathbf{H^Q} = \text{AudioEncoder}(\mathbf{Y}) \tag{3}$$

D. PHONEME ENCODER

The phoneme encoder uses a Bi-LSTM network. We first embed the reference phoneme sequence into the phoneme with the embedding dimension of 512 and then feed the phoneme embedding into the Bi-LSTM encoder with the input dimension of 512; the size of the hidden layer is 128. We mark the input reference phoneme sequence as $\mathbf{P} = [\mathbf{p_1}, \ldots \mathbf{p_n}]$. Our phoneme encoder can be expressed as

$$\mathbf{h^K}, \mathbf{h^V} = \text{PhonemeEncoder}(\mathbf{P}) \tag{4}$$

E. FEATURE FUSION

Our feature fusion method uses the dot-product attention mechanism [34], where $\mathbf{h^Q}$, $\mathbf{h^K}$, and $\mathbf{h^V}$ are used as query vectors, key vectors, and value vectors for attention calculation. Then, we use the dot-product attention mechanism to input $\mathbf{h^Q}$, $\mathbf{h^K}$, and $\mathbf{h^V}$, and calculate the final fusion feature sequence $\mathbf{M} = [\mathbf{m_1}, \ldots \mathbf{m_n}]$. The calculation process of

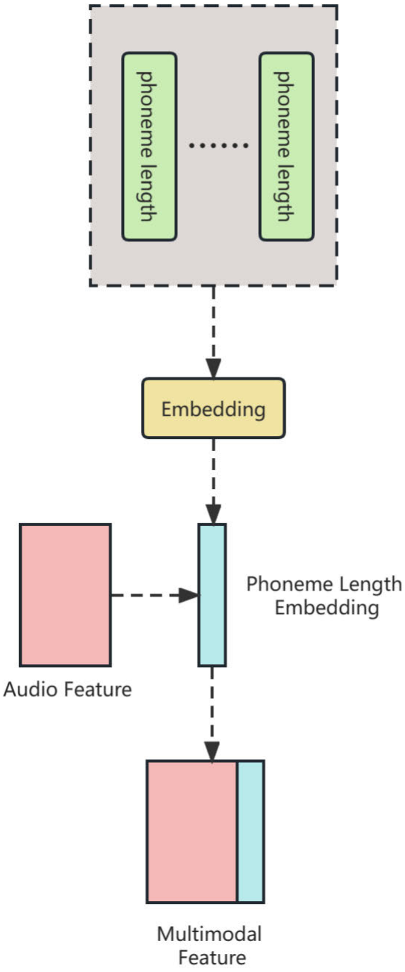

FIGURE 3. Multi-modal information fusion.

dot-product attention can be expressed as follows:

$$\mathbf{M} = \text{Concat}(\mathbf{h}^{\mathbf{Q}}, \mathbf{C}) \tag{5}$$

$$\mathbf{C} = \text{Attention}(\mathbf{h}^{\mathbf{Q}}, \mathbf{h}^{\mathbf{K}}, \mathbf{h}^{\mathbf{V}}) \tag{6}$$

$$\text{Attention}(\mathbf{h}^{\mathbf{Q}}, \mathbf{h}^{\mathbf{K}}, \mathbf{h}^{\mathbf{V}}) = \alpha \cdot \mathbf{h}^{\mathbf{V}} \tag{7}$$

$$\alpha = \frac{\exp(\text{score}(\mathbf{h}^{\mathbf{K}}, \mathbf{h}^{\mathbf{Q}}))}{\Sigma\exp\left(\text{score}(\mathbf{h}^{\mathbf{K}}, \mathbf{h}^{\mathbf{Q}})\right)} \tag{8}$$

$$\text{score}(\mathbf{h}^{\mathbf{K}}, \mathbf{h}^{\mathbf{Q}}) = \mathbf{h}^{\mathbf{K}} \cdot \mathbf{h}^{\mathbf{Q}} \tag{9}$$

where $\mathbf{C}$ is the vector for frame-level alignment via the attention mechanism, α is the attention weight, and the score is $\mathbf{h}^{\mathbf{K}} \cdot \mathbf{h}^{\mathbf{Q}}$. Then, the hidden state sequence $\mathbf{h}^{\mathbf{Q}}$ is output after audio encoding and the hidden state sequence $\mathbf{M}$, obtained by the dot-product attention mechanism, is spliced into the linear layer for down-sampling to obtain the final fused feature hidden state sequence. Here, the purpose of down-sampling is to reduce the number of dimensions of the spliced features, reduce the amount of computation in the decoding stage, and speed up the decoding process. The feature fusion process is shown in Fig.4.

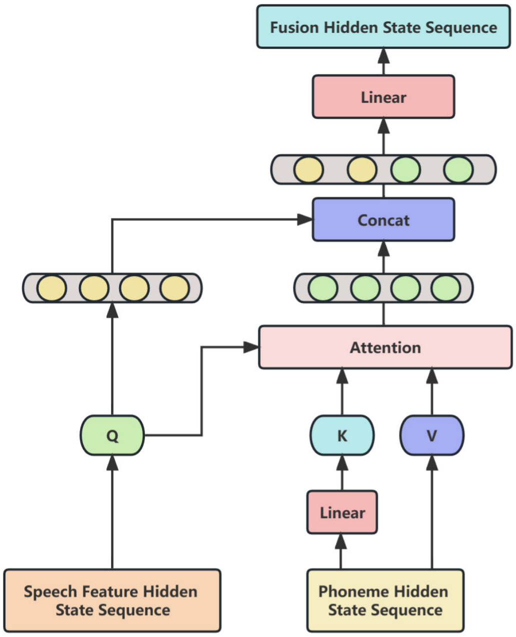
Fusion Hidden State Sequence
Linear
Concat
Attention
Q
K
V
Linear
Speech Feature Hidden State Sequence
Phoneme Hidden State Sequence

FIGURE 4. Feature fusion based on attention.

F. DECODER

The performance following the joint training of CTC and the attention mechanism is greatly improved. In the training process, the main operation is to use the output of the encoder as the input of the decoder and the input of CTC, respectively. Our decoder uses the Transformer structure, and the hidden state sequence obtained by the encoder calculates the joint loss through CTC and attention and uses the joint loss as the final loss value for the parameters of the back-propagation optimization model. We mark the loss calculated by CTC as L_{ctc}, and the loss calculated by attention is marked as L_{att}. Our joint loss L_{loss} can be expressed as

$$L_{\text{loss}} = \lambda L_{\text{ctc}} + (1-\lambda)L_{\text{att}} \tag{10}$$

where λ is the weight value assigned to CTC loss, and $(1-\lambda)$ is the weight value assigned to attention loss.

G. SECONDARY DECODING

Secondary decoding mainly uses the decoder to decode the beam search [35] results twice, thus changing the candidate ranking of the whole-sentence results. At each time step, beam search saves the top n candidate sequences and predicts the next phoneme for each candidate sequence. If there are k phonemes in the phoneme set, k prediction results will be generated. Then, from the resulting nk new sequences, the top n sequences are selected as the candidate sequences.The secondary decoding process obtains n of the best results from the output of the decoder through beam search, and then inputs the n best results as the label of the decoder again, which solves the problem of the decoder having no prior knowledge in the first decoding stage. In the experiment,

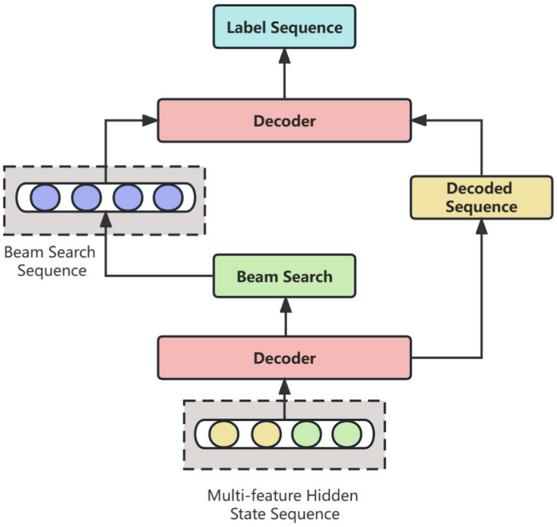

FIGURE 5. Secondary decoding process based on beam search.

we set n to 10. We express the encoded multi-feature hidden state vector as $\mathbf{M} = [\mathbf{m_1}, \ldots \mathbf{m_n}]$, $\mathbf{M}$ here is equivalent to the vector in Formula 5.Then, the whole secondary decoding process can be expressed by the following formula:

$$\text{decoder}_{\text{out}} = \text{Decoder}(\mathbf{M}) \tag{11}$$

$$\text{hyps}_{\text{ctc}} = \text{BeamSearch}(\text{decoder}_{\text{out}}) \tag{12}$$

$$\text{hyps}_{\text{att}} = \text{Decoder}(\text{hyps}_{\text{ctc}}, \text{decoder}_{\text{out}}) \tag{13}$$

where hyps_{ctc} represents the first decoded phoneme tag sequence. We decode the multi-feature hidden state vector at the decoder end to obtain the corresponding output sequence through model decoding and then calculate the predicted phoneme tag sequence hyps_{ctc} through beam search, and then the predicted phoneme tag sequence hyps_{ctc} and the output of the CTC and decoder are sent to the Transformer decoder for secondary decoding to obtain the final predicted phoneme tag sequence hyps_{att}. The whole secondary decoding process is shown in Fig.5.

IV. SPEECH CORPUS

A. DATA CREATION PROCESS

To be consistent with our baseline model, we also use the dataset PSC-Reading [14], built by Shen et al., as the source of experimental data for this paper. The recording texts of the dataset are all from the Mandarin Proficiency Test question bank, which is to ensure the consistency between the self-built dataset and the real test data of the Mandarin Proficiency Test. In this question bank, there are 60 short-text reading questions, ten of which were selected as reference texts for the data recordings, each with a word count between 400 and 600 words.

In the process of building the data, dozens of university or graduate students were recruited to record the audio in a quiet environment. Speakers were recorded reading aloud from the selected recorded text. The recording equipment used a headset connected to an office computer, and the recording software used Praat with a sampling rate of 44.1 kHz, mono, and saved in Flac format. Afterward, each recorded text was checked manually to screen out data with too many mispronunciations or with too much outside noise, and the speakers were organized for a secondary collection of data. Each speaker was assigned ten short texts to read aloud, and the speaker recorded from the complete text so that each voice included chapter-level short-text readings. Because the input for phoneme recognition model training is typically short audio of fewer than 20 s, in the later annotation stage, the original long audio was segmented into short audio based on sentences through forced alignment. Silent portions between adjacent sentences longer than 1 s were removed and all 10 spoken texts were divided into a total of 240 short sentences based on the sentence boundary, with an average length of 20.65 syllables per short sentence.

B. DATA ANNOTATION SPECIFICATION

This section describes the method of annotation of PSC-Reading using Praat. To improve the annotation efficiency, manual fine annotation was performed on top of the machine's coarse annotation TextGrid files generated by forced alignment. For chapter-level article audio (between 150 and 200 s in length), it was manually divided into short audio of approximately 10 s in length after the sentences were manually divided using the CTC-Segmentation [36] tool provided by the ESPNet2 framework to force alignment by sentence. The short audio obtained from the segmentation was kept silent for no more than 0.5 s before and after. Using the speech evaluation interfaces provided by Unisound, IFLYTEK, etc., we obtained the machine scores and generated the coarse standard TextGrid according to the annotation specification. The TextGrid annotation file uses UTF-8 encoding, and when using Hanyu Pinyin for annotation, the tones are represented by numbers, 1-4 for one to four tones, respectively, 5 for light tones, and other numbers represent special tones that do not exist in standard Mandarin. When using "v" for "ü" in phoneme annotation, in the syllable layer and the pinyin layer, "ü", which is represented as "u" according to the Hanyu Pinyin standard, does not need to be changed; in the phoneme layer, the phoneme containing ü should first be reduced to its original form and then represented as "v", e.g., "jun1" is first broken down into the phoneme "j+ün1" and then into "j+vn1".

For the Erhua phenomenon, the Erhua part of "er" is separated from the phoneme as a syllable and is marked at the word level together without a tone value. For the zero consonants "y" and "w", a special rhyme notation is used to replace their position in the phoneme notation as a pseudo vowel. The annotation includes the following hierarchy:

1) Sentence text, the entire sentence and its corresponding time range, presented in Chinese characters;
2) Pinyin of sentences, including the pinyin of the entire sentence;

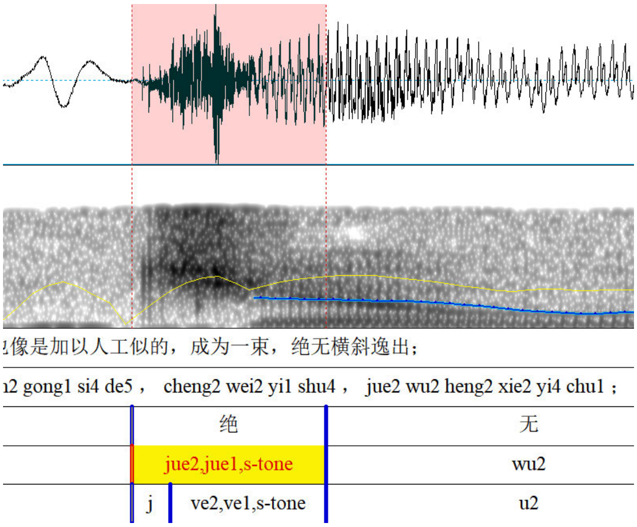

FIGURE 6. Mispronunciation of tones.

TABLE 1. Special phonetic transcriptions.

Pinyin	Consonant	Vowel
zhi1	zh	ix1
chi1	ch	ix1
shi1	sh	ix1
zi1	z	iy1
ci1	c	iy1
si1	s	iy1
ri1	r	iz1

3) Words, that is, each word and its corresponding time range, are labeled with Chinese characters;
4) Syllables, the time range of each syllable in pinyin and silence, using pinyin notation;
5) Phonemes, the time range of each phoneme and silence, using pinyin notation;

Among them, words, syllables, and phonemes need to be marked with three types of mispronunciation—-substitution, insertion, and deletion—in the format of expected correct pronunciation, actual pronunciation, and error type. For example, "jue2, jue1, s-tone" represents vowel tone mispronunciation. "jue2" is the pinyin of the corresponding Chinese character, and "jue2, jue1, s-tone " means that the speaker mispronounced "jue2" into "jue1". We use "s-tone" to indicate a tone error. See the following for specific annotation specifications, and an example of an annotation is shown in Fig.6.

For special phonemes, we perform the corresponding phoneme transcriptions during the annotation process, such as the transcription of some vowels (zh, ch, sh, z, c, s, and r) with i. The transcription rules are shown in Table 1.

After completing the PSC-Reading data annotation, several trained graduate students were organized as data proofreaders to finely proofread the data. After proofreading to ensure that all data are error-free, the data are organized into the format of the L2-Arctic dataset [37], and the construction of the PSC-Reading-24 dataset is completed.

TABLE 2. Details of the PSC-Reading dataset.

	PSC-Reading-G1	PSC-Reading-24		
	train	train	dev	test
Speakers	5	12	6	6
Utterance	1200	2864	1435	1434
Phonemes	45,846	109,821	54,886	55,014
Duration(/h)	2.41	4.15	2.24	2.20

TABLE 3. Phoneme error details of the PSC-Reading dataset.

	Corr.	Sub.	Ins.	Del.	Total
train	153,416	1521	730	952	155,889
dev	54,064	762	60	205	55,031
test	54,187	556	271	291	55,034

C. DATA STATISTICS

The PSC-Reading-24 dataset includes a total of 24 speakers with a male-to-female gender ratio of 12:12. 240 raw audio recordings were recorded and cut into 5733 utterances, and a small number of recorded utterances that did not meet the recording criteria were discarded before annotation. The total length of the dataset was 8.59 h, with an average sentence length of 5.4 s. As a comparison, L2-Arctic, a commonly used public dataset for English second language acquisition in academia, includes 24 speakers, of whom each has 150 recordings annotated by experts for mispronunciation, with a total of 3.66 h of annotated data in total. We followed the L2-Arctic and TIMIT [38] data and used the above data annotation format to additionally annotate the standard example audio of the first 10 sets of Mandarin read-aloud texts from five different training institutions or Mandarin textbooks, notated as PSC-Reading-G1.

The PSC-Reading-G1 and PSC-Reading-24 training sets were used as the final training set with a total training time of 6.56 h to enhance the model's ability to detect substandard pronunciation and finally build up the final dataset for detecting and diagnosing mispronunciation in read-aloud questions of the Mandarin Proficiency Test. The statistical information of the dataset is shown in Table 2, and the statistical information of the number of mispronunciations in the dataset is shown in Table 3.

V. EXPERIMENTS

A. EXPERIMENTAL ENVIRONMENT

The experiment in this paper was carried out on an x86 server of Ubuntu 20.04 system, and the GPU used was NVIDIA A40. The in-depth learning framework used in the experiment was PyTorch 1.13.1, the CUDA runtime version was 11.6, the Python runtime version used was Python 3.8.16, and the CTC loss function used was the PyTorch built-in implementation. In this paper, feature extraction, phoneme model building and training were implemented using the WeNet [39], [40] framework, and the calculation of each index of mispronunciation detection and diagnosis was conducted using the method given by Fu et al. [25].

The batch size parameter was set to 32 for model training in the experiments of this paper, the WavLM-Transformer [14] and CNN-RNN-CTC [10] models based on the beam search algorithm were used as the baseline model, the modeling unit was the phoneme, and the word list size (number of tag class) was 188.

In this paper, the speed perturbation data enhancement method was used in the model training process. The range of speech speed conversion used in this paper was 0.9, 1.0, and 1.1 times.

B. MISPRONUNCIATION DETECTION AND DIAGNOSIS

We used the phoneme error rate (PER), F1 score, detection accuracy and diagnosis accuracy as the performance indicators of the model.

$$\text{PER} = \frac{(S+D+I)}{N} \tag{14}$$

$$\text{DetectionAccuracy} = \frac{(TA+TR)}{(TA+FR+FA+TR)} \tag{15}$$

$$\text{DiagnosisAccuracy} = \frac{CD}{(CD+DE)} \tag{16}$$

where S, D, and I are the number of substitution, deletion, and insertion errors, and N is the total number of phonemes in the reference phoneme sequence. Before calculating the F1 score, it is necessary to divide the mispronunciation detection results of the model into four cases: true accept (TA), true rejection (TR), false accept (FA), and false rejection (FR). Then, the F1 score can be calculated according to the number of four types of result samples. The mispronunciation detection results of the true rejection (TR) type can be further divided into two categories according to whether the mispronunciation type is correctly determined: correct diagnosis (CD) and diagnosis error (DE). The error detection and diagnosis performance of the model can be comprehensively measured by combining F1, detection accuracy, and diagnosis accuracy.

We use the WavLM-Transformer [14] and CNN-RNN-CTC [10] model as the baseline model for our experiments, and we also conduct ablation experiments for models using different encoders, with or without multi-features and multi-modality, and with or without secondary decoding in our experiments. Considering the impact of different CTC loss weights on the performance of the model when using CTC/Attention joint loss, we conducted a detailed comparative experiment on the performance of the model under different CTC loss weights. We adjusted the CTC loss weight from 0 to 1. When the CTC loss weight is 0, it means that only a single Attention loss is used in the model training process. When the CTC loss weight is 1, it means that only a single CTC loss is used in the model training process. CTC loss weight between 0 and 1 means that the model uses CTC/Attention joint loss during the training process. Table 4 shows the experimental results under different CTC loss weights. It can be seen from the experimental results that the model achieved the best F1 value and PER on the test set when the CTC loss weight was set to 0.6. Table 5 compares the MDD results under different CTC loss weights. It can be seen from the table that the impact of different CTC loss weights on MDD is uneven. When the CTC loss weight is 0.4, the model achieved the best diagnostic accuracy on the test set, while when the CTC loss weight was set to 1.0, the model achieved the best detection accuracy on the test set. Therefore, considering all the indicators of pronunciation error detection and diagnosis, we finally set the CTC loss weight to 0.6.The distribution of F1 score and PER index obtained by the model under different CTC loss function weights λ are shown in Fig.7.According to the experimental results in Table 6, it can be seen that the multi-feature, multi-modal mispronunciation detection and diagnosis model based on the Squeezeformer encoder has the best PER and F1 value in a series of comparative model experiments after secondary decoding. It achieved an F1 score of 0.7943 and a PER of 1.50%. Relative to the WavLM-Transformer model using the beam search algorithm, the F1 was relatively improved by 0.3883, and the PER was reduced by 2.66%. From the experimental results, it can be seen that the model using Squeezeformer as the encoder has a significant performance improvement compared with the model using Transformer as the encoder. Taking the Squeezeformer encoder as an example, after adding multi-modal information based on the use of feature fusion and secondary decoding, the F1 value is improved by 0.0275, the PER is reduced by 0.12%, and the diagnostic accuracy is improved by 0.59%; thus, we can verify that the model has a certain performance improvement after incorporating multi-modal information. Among the secondary decoding algorithms, we use SD (secondary decoding) as representative. From the experimental results, we can see that the SD decoding method shows better performance compared with decoding using the beam search method using the same encoder. We also take the Squeezeformer encoder as an

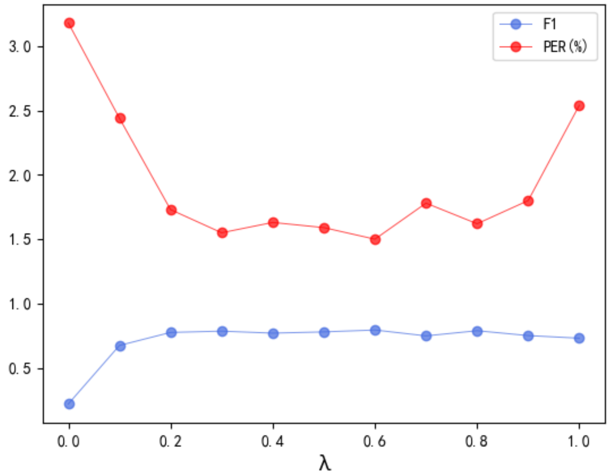

FIGURE 7. F1 value and PER index under different λ.

TABLE 4. Experimental results under different CTC weight.

λ(CTC Weight)	Recall	Precision	F-measure(%)	PER(%)
0	0.2182	0.2218	22.00	3.48
0.1	0.8810	0.5463	67.44	2.44
0.2	0.8998	0.6816	77.56	1.73
0.3	0.8882	0.7043	78.56	1.55
0.4	0.8837	0.6823	77.01	1.63
0.5	0.8784	0.7009	77.97	1.59
0.6	0.8909	**0.7165**	**79.43**	**1.50**
0.7	0.8605	0.6630	74.89	1.78
0.8	0.8801	0.7146	78.88	1.62
0.9	0.8676	0.6621	75.11	1.80
1.0	**0.9141**	0.6087	73.08	2.54

Note: Bold values represent the best results.

TABLE 5. MDD results under different CTC weight.

λ(CTC Weight)	canonicals		mispronunciations	Accuracy(%)
	False Accept	False Rejection	True Rejection	
			Correct Diag	
0	78.18%	15.8%	54.92%	65.78%
0.1	11.90%	1.51%	87.31%	93.27%
0.2	10.02%	0.87%	86.88%	94.47%
0.3	11.18%	0.77%	89.02%	94.03%
0.4	11.63%	0.85%	**89.47%**	93.80%
0.5	12.16%	0.77%	87.37%	93.52%
0.6	10.91%	**0.73%**	88.45%	94.16%
0.7	13.95%	0.90%	87.94%	92.64%
0.8	11.99%	**0.73%**	88.01%	93.64%
0.9	13.24%	0.91%	87.42%	92.95%
1.0	**8.59%**	1.21%	86.40%	**94.97%**

Note: Bold values represent the best results.

TABLE 6. Experimental results.

Models	Recall	Precision	F-measure(%)	PER(%)
Based				
CNN-RNN-CTC	0.6750	0.2245	33.70	6.07
Transformer	0.8050	0.1558	26.11	10.76
WavLM-Transformer	0.6458	0.2960	40.60	4.16
Ours				
Squeezeformer	0.7236	0.2361	35.60	6.24
Squeezeformer-SD	0.7066	0.2506	37.00	5.53
Squeezeformer-PhnDep	0.9249	0.6548	74.53	2.13
Squeezeformer-PhnDep-Multi	**0.9365**	0.6556	77.13	2.06
Squeezeformer-PhnDep-SD	0.8766	0.6941	76.68	1.62
Squeezeformer-PhnDep-Multi-SD	0.8909	**0.7165**	**79.43**	**1.50**

Note: "Based" refers to our baseline, "Ours" refers to our work,"PhnDep" refers to feature fusion, "Multi" refers to multi-modality, and "SD" refers to secondary decoding. The baseline corresponding to "SD" is the beam search algorithm.Bold values represent the best results.

example. Under the condition of using multiple features and multiple modes, the decoding process using the SD decoding method improves by 2.3% in terms of F1 value, 0.56% in terms of PER reduction, and 1.34% in terms of diagnostic accuracy compared with using the beam search decoding method. Table 7 demonstrates the performance of our model in terms of correct diagnosis and detection accuracy, and it can be seen from the experiments that the Squeezeformer encoder-based multi-feature and multi-modal approach has the highest diagnosis accuracy after secondary decoding, with a diagnostic accuracy of 88.45%. The multi-feature model based on the Squeezeformer encoder achieves 96.19% detection accuracy using beam search combined with multi-modal methods. Overall, in this experiment, the model based on the Squeezeformer encoder and using feature fusion and multi-modal information, which uses a secondary decoding

TABLE 7. MDD results.

Models	canonicals		mispronunciations	Accuracy(%)
	False Accept	False Rejection	True Rejection	
			Correct Diag	
Based				
CNN-RNN-CTC	32.50%	4.82%	67.99%	94.62
Transformer	19.50%	9.00%	71.00%	85.04
WavLM-Transformer	35.42%	3.17%	83.93%	96.18
Ours				
Squeezeformer	27.64%	4.83%	73.92%	83.89
Squeezeformer-SD	29.34%	4.36%	74.94%	83.50
Squeezeformer-PhnDep	7.51%	1.01%	88.01%	95.64
Squeezeformer-PhnDep-Multi	**6.35%**	1.02%	87.11%	**96.19**
Squeezeformer-PhnDep-SD	12.34%	0.80%	87.86%	93.44
Squeezeformer-PhnDep-Multi-SD	10.91%	**0.73%**	**88.45%**	94.16

Note: “Based” refers to our baseline, “Ours” refers to our work,“PhnDep” refers to feature fusion, “Multi” refers to multi-modality, and “SD” refers to secondary decoding. The baseline corresponding to “SD” is the beam search algorithm.Bold values represent the best results.

mechanism in the decoding process, performs better than the baseline model.

VI. CONCLUSION

Combined with the more advanced Squeezeformer model in speech recognition tasks, this paper proposes a reference text-related mispronunciation detection and diagnosis framework based on Squeezeformer, dual encoders, multi-modal features, and the secondary decoding mechanism. This model can effectively combine multi-modal information to improve the representation ability of speech features, so that the model can further learn the intrinsic characteristics of the speaker’s pronunciation content, thus improving the accuracy of the model phoneme recognition and the quality of mispronunciation detection and diagnosis. From a series of comparative experiments, it can be seen that in the mispronunciation detection and diagnosis dataset, the model with the Squeezeformer encoder incorporating multi-modal information and using the secondary decoding mode has better performance. Compared with the baseline model, our model has better performance in terms of the F1 value, PER, diagnostic accuracy, and detection accuracy.

Through the above research, our model can be applied to computer-assisted language learning (CALL). This research has important theoretical and practical significance in the field of mispronunciation detection and diagnosis., being able to theoretically improve the performance of mispronunciation detection and diagnosis, reduce the errors in model performance, and more effectively complete mispronunciation detection and diagnosis tasks. At the same time, this study also provides important theoretical information for further research in the field of mispronunciation detection and diagnosis. In addition, this research also has great potential value in the field of education, to help people better learn Mandarin and provide a series of feedback on pronunciation quality. At the same time, it will greatly reduce the pressure of manual evaluation.

To improve the performance of the Mandarin mispronunciation detection and diagnosis model, we will continue to explore the multi-task learning method of integrating data scoring tags into the multi-feature aspect and adding scoring tasks based on mispronunciation detection and diagnosis, in order to further improve the mispronunciation detection and diagnosis performance.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[2] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented transformer for speech recognition,” 2020, *arXiv:2005.08100*.

[3] S. Kim, A. Gholami, A. Shaw, N. Lee, K. Mangalam, J. Malik, M. W. Mahoney, and K. Keutzer, “Squeezeformer: An efficient transformer for automatic speech recognition,” 2022, *arXiv:2206.00888*.

[4] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, “Speech recognition using deep neural networks: A systematic review,” *IEEE Access*, vol. 7, pp. 19143–19165, 2019.

[5] K. Chowdhary and K. Chowdhary, “Natural language processing,” *Fundam. Artif. Intell.*, pp. 603–649, 2020.

[6] K. Shim and W. Sung, “A comparison of transformer, convolutional, and recurrent neural networks on phoneme recognition,” 2022, *arXiv:2210.00367*.

[7] V. Likic, “The Needleman–Wunsch algorithm for sequence alignment,” Lect. Given at 7th Melbourne Bioinf. Course, Bi021 Mol. Sci. Biotechnol. Inst., Univ. Melbourne, Tech. Rep., 2008, pp. 1–46.

[8] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, “Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers,” *Speech Commun.*, vol. 67, pp. 154–166, Mar. 2015.

[9] A. M. Harrison, W.-K. Lo, X.-J. Qian, and H. Meng, “Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training,” in *Proc. Int. Workshop Speech Lang. Technol. Educ.*, 2009.

[10] W. Leung, X. Liu, and H. Meng, “CNN-RNN-CTC based end-to-end mispronunciation detection and diagnosis,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 8132–8136.

[11] K. Kawakami, “Supervised sequence labelling with recurrent neural networks,” Ph.D. dissertation, Tech. Univ. Munich, Munich, Germany, 2008.

[12] J. Salazar, K. Kirchhoff, and Z. Huang, “Self-attention networks for connectionist temporal classification in speech recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 7115–7119.

[13] Y. Feng, G. Fu, Q. Chen, and K. Chen, “SED-MDD: Towards sentence dependent End-To-End mispronunciation detection and diagnosis,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3492–3496.

[14] Y. Shen, Q. Liu, Z. Fan, J. Liu, and A. Wumaier, "Self-supervised pre-trained speech representation based end-to-end mispronunciation detection and diagnosis of Mandarin," *IEEE Access*, vol. 10, pp. 106451–106462, 2022.

[15] A. Graves and A. Graves, "Long short-term memory," in *Supervised Sequence Labelling With Recurrent Neural Networks*, 2012, pp. 37–45.

[16] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 3, pp. 288–298, Mar. 2001.

[17] D. Su, X. Wu, and L. Xu, "GMM-HMM acoustic model training by a two level procedure with Gaussian components determined by automatic model selection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 4890–4893.

[18] G. D. Forney Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[19] C. A. Floudas and V. Visweswaran, "A global optimization algorithm (GOP) for certain classes of nonconvex NLPs—I. Theory," *Comput. Chem. Eng.*, vol. 14, no. 12, pp. 1397–1417, Dec. 1990.

[20] J. Novoa, J. Wuth, J. P. Escudero, J. Fredes, R. Mahu, and N. B. Yoma, "DNN-HMM based automatic speech recognition for HRI scenarios," in *Proc. 13th ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, Mar. 2018, pp. 150–159.

[21] J. Shi, N. Huo, and Q. Jin, "Context-aware goodness of pronunciation for computer-assisted pronunciation training," 2020, *arXiv:2008.08647*.

[22] K. Li, X. Qian, and H. Meng, "Mispronunciation detection and diagnosis in l2 English speech using multidistribution deep neural networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 1, pp. 193–207, Jan. 2017.

[23] D. Amodei et al., "Deep speech 2: End-to-end speech recognition in English and mandarin," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 173–182.

[24] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1240–1253, Dec. 2017.

[25] K. Fu, J. Lin, D. Ke, Y. Xie, J. Zhang, and B. Lin, "A full text-dependent end to end mispronunciation detection and diagnosis with easy data augmentation techniques," 2021, *arXiv:2104.08428*.

[26] Z. Zhang, Y. Wang, and J. Yang, "Text-conditioned transformer for automatic pronunciation error detection," *Speech Commun.*, vol. 130, pp. 55–63, Jun. 2021.

[27] S. Nadig, V. Ramasubramanian, and S. Rao, "Multi-target hybrid CTC-attentional decoder for joint phoneme-grapheme recognition," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jul. 2020, pp. 1–5.

[28] D. Kenny, "Machine translation," in *Routledge Encyclopedia of Translation Studies*. Evanston, IL, USA: Routledge, 2019, pp. 305–310.

[29] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, *arXiv:1803.02155*.

[30] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," 2017, *arXiv:1707.05589*.

[31] Y. Pang, M. Sun, X. Jiang, and X. Li, "Convolution in convolution for network in network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1587–1597, May 2018.

[32] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021.

[33] M. India, P. Safari, and J. Hernando, "Self multi-head attention for speaker recognition," 2019, *arXiv:1906.09890*.

[34] S. Zhuoran, Z. Mingyuan, Z. Haiyu, Y. Shuai, and L. Hongsheng, "Efficient attention: Attention with linear complexities," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3531–3539.

[35] A. Kumar, S. Vembu, A. K. Menon, and C. Elkan, "Beam search algorithms for multilabel learning," *Mach. Learn.*, vol. 92, no. 1, pp. 65–89, Jul. 2013.

[36] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *Proc. 22nd Int. Conf. Speech Comput. (SPECOM)*, St. Petersburg, Russia: Springer, Oct. 2020, pp. 267–278.

[37] G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "L2-ARCTIC: A non-native English speech corpus," in *Proc. INTERSPEECH*, Sep. 2018, pp. 2783–2787.

[38] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus CD-ROM. Nist speech disc 1-1.1," NASA STI/Recon Tech. Rep., 1993, p. 27403, vol. 93.

[39] Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, and X. Lei, "WeNet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," 2021, *arXiv:2102.01547*.

[40] B. Zhang, D. Wu, Z. Peng, X. Song, Z. Yao, H. Lv, L. Xie, C. Yang, F. Pan, and J. Niu, "WeNet 2.0: More productive end-to-end speech recognition toolkit," 2022, *arXiv:2203.15455*.

SHEN GUO received the B.Eng. degree in software engineering from the Luoyang Institute of Technology. He is currently pursuing the master's degree in computer technology with Xinjiang University. His research interest includes mispronunciation detection and diagnosis(MDD).

ZAOKERE KADEER is currently an Experimentalist with Xinjiang University. Her research interest includes natural language processing.

AISHAN WUMAIER (Member, IEEE) received the Ph.D. degree in computer applied technology from Xinjiang University. He is currently a Professor with Xinjiang University, where he is the Ph.D. Supervisor. His research interests include sentiment analysis, machine translation (MT), and mispronunciation detection and diagnosis (MDD).

LIEJUN WANG received the Ph.D. degree. He is currently a Professor with Xinjiang University. He is the Ph.D. Supervisor. His main research interests include video communication processing, image recognition and processing, wireless sensor networks, and security.

CONG FAN received the B.Eng. degree in software engineering from the Hunan Institute of Information Technology. She is currently pursuing the master's degree in computer technology with Xinjiang University. Her research interest includes off-topic detection.